A Keyword-based Monolingual Sentence Aligner in Text Simplification

[1]Chung-Chi Huang [2]Maxine Eskenazi [3]Mei-Hua Chen [4]Ping-Che Yang
[1,2]Language Technologies Institute, CMU, United States
[3]Hua Fan University, Taipei, Taiwan
[4]Institute for Information Industry, Taipei, Taiwan
{[1]u901571,[3]chen.meihua}@gmail.com,[2]max+@cs.cmu.edu,[4]maciaclark@iii.org.tw

Abstract

We introduce a method for learning to align sentences in monolingual parallel articles for text simplification. In our approach, word keyness is integrated to prefer aligning essential words in sentences. The method involves estimating word keyness based on TF*IDF and semantic PageRank, and word nodes' parts-of-speech and degrees of reference. At run-time, the keyword analyses are used as word weights in sentence similarity measure. And a global dynamic programming goes through sentence similarities further weighted by aligned content-word ratios and positions of aligned words to determine the optimal candidates of parallel sentences. We present a prototype sentence aligner, *KEA*, that applies the method to monolingual parallel articles. Evaluation shows that *KEA* pays more attention to key words during sentence aligning and outperforms the current state-of-the-art in alignment accuracy and f-measure. Our pilot study also indicates that language learners benefit from our sentence-aligned parallel articles in reading comprehension test.

1 Introduction

Many articles are posted on the Web every day, and an increasing number of educational websites specifically provide articles for audiences with different needs. For example, NewsInLevels (www.newsinlevels.com) and BreakingNewsEnglish (www.breakingnewsenglish.com) select news articles and provide versions with different readability for language learners. Simple Wikipedia (simple.wikipedia.org) and EasierEnglishWiki (eewiki.newint.org) contain articles easier to read with simpler vocabulary and syntactic structure than English Wikipedia and New Internationalist for people with low literacy. And SoundReading (www.soundreading.com) even has audio recording for those with learning disabilities such as dyslexia.

Language learning websites such as NewsInLevels and EasierEnglishWiki typically simplify original articles into easier ones and present the original and easier articles as pairs to non-native speakers, children, or lay people. However, language learners may want to compare the article pairs conveying the same information at sentence level, and most text simplification systems build on top of original and simplified sentence pairs. Unfortunately, current monolingual sentence alignment methods treat article sentences as bags of words, equally weight words, and align sentences with high word-overlap ratios. These article pairs could be sentence aligned more accurately if a system distinguished words of different importance and leveraged their importance levels in articles while aligning.

Consider the original-simplified article pair in Figure 1. The best sentence alignment methods are probably not the ones with equal word weights (i.e., weights are the same with "*the*" and "*gorilla*" and the same with "*everything*" and "*project*"). A

28th Pacific Asia Conference on Language, Information and Computation pages 542–550

Original article:
(1) An army of gorillas came to the River Thames! (2) They aren't real animals, but statues dressed up as people. (3) There are 20 life-sized, individually decorated gorillas, including Elvis and Spiderman. (4) They are here to bring people's attention to the problems of gorillas. (5) There are very few gorillas in the wild and their number decreases every year. (6) People destroy their homes and kill them. (7) But not everything is lost. (8) There are a lot of excellent projects, which can help gorillas. (9) These statues are one of such projects. (10) The show will finish on 22 September.

Simplified article:
(s1) There are gorillas in London. (s2) They are not animals. (s3) They are big statues. (s4) There are 20 gorillas in London. (s5) They have different clothes. (s6) There is Elvis or Spiderman. (s7) People made them. (s8) They want to show problems of gorillas. (s9) A lot of gorillas die every year. (s10) People take their homes. (s11) They kill them too. (s12) But there are a lot of projects. (s13) These projects can help gorillas. (s14) The statues are one of these projects. (s15) The show will finish on 22 September.

Words' keyness scores:
gorilla:0.15; September:0.13; project:0.10; statue:0.09; Spiderman: 0.08; Elvis: 0.06; 22: 0.06; home:0.06; people:0.05; problem:0.05; animal: 0.04; show: 0.04 …

Sentence pairs:
(1,s1), (2,s2), (2,s3), (3,s4), (3,s6), (4,s8), (5,s9), (6,s10), (6,s11), (8,s12), (8,s13), (9,s14), (10,s15)

Figure 1. An example *KEA* sentence alignment for an article pair.

good aligning approach might take into account the words' significance in the pair. Intuitively, word significance can be evaluated by keyword extraction methods and by leveraging word significance, sentence aligners can be biased towards aligning sentences with more words that are more essential.

We present a new system, *KEA* (keyword extraction based sentence aligner), that automatically learns to align sentences, considering word keyness, of monolingual parallel articles. That is, *KEA* aligns texts in the same language at sentence level that are "translation" of each other with different readability. An example *KEA* sentence alignment for an article pair is shown in Figure 1. *KEA* has determined the keyness scores of the words in the article pair. *KEA* learns these scores automatically during training by using TF*IDF and PageRank with semantic information (see details in Section 2). Both are famous keyword extraction methods.

At run-time, *KEA* starts with a pair of monolingual parallel articles. *KEA* then computes similarity scores among sentences in the original and simplified article based on words' keyness scores from TF*IDF and PageRank. Cosine similarity is adopted to evaluate sentence-wise similarity with the help of alignment ratio of content words and differences of relative aligned word positions. Based on sentence-level similarity, *KEA* employs global dynamic programming with deletion and insertion operation to generate the optimal sentence alignment for the pair. In our prototype, *KEA* returns sentence pairs for evaluation and language learning directly (see Figure 1); alternatively, the sentence pairs returned by *KEA* can be used as input to a text simplification system.

2 The KEA System

Submitting monolingual parallel articles to sentence aligners counting word overlaps often does not work very well. Such aligners typically assign equal weights to words. Unfortunately, some words (e.g., content words) are more important than others (e.g., function words) and aligners should pay more attention to topic/key words while sentence aligning. To align monolingual parallel articles at sentence level, a promising approach is to automatically integrate words' keyness that reflects the significance of words in the articles.

2.1 Problem Statement

We focus on the first step of automated text simplification: aligning monolingual parallel articles at sentence level. These sentence pairs are

then returned as the output of the system. The returned sentences pairs can be examined for alignment accuracy, used for language learning, or passed on to sophisticated text simplification models (e.g., (Zhu et al., 2010) and (Woodsend and Lapata, 2011)). Thus, it is crucial that the aligned sentences be accurate. At the same time, the set of identified sentence pairs cannot be so small that it bores the user or hurts the performance of the subsequent (typically data intensive) simplification models. Therefore, our goal is to return a reasonable-sized set of parallel sentences that, at the same time, must contain correct sentence mappings of the parallel articles. We now formally state the problem that we are addressing.

Problem Statement: We are given a monolingual parallel article pair, specifically, an original article Art_o and its simplified counterpart Art_s. Our goal is to retrieve a set of sentence pairs that are likely to be the parallel sentences between Art_o and Art_s. For this, we transform Art_o and Art_s into a set of sentences, $Sent_{o,1}$,…, $Sent_{o,m}$, $Sent_{s,1}$,…, $Sent_{s,n}$, and calculate keyness scores for words within such that the sentences are aligned considering word importance and the candidate set of sentence pairs are likely to contain parallel sentences in Art_o and Art_s.

In the rest of this section, we describe our solution to this problem. First, we define a strategy for distinguishing words of different importance in the parallel articles and assigning them keyness socres accordingly (Section 2.2). This strategy relies on TF*IDF and PageRank. In this section, we also describe how we extend PageRank to semantic one using semantic information such as keyword preference model, and words' parts-of-speech and degrees of reference in the articles. Finally, we show how *KEA* applies global dynamic programming to align sentences at run-time by leveraging keyness scores for words, and sentence-level ratios of aligned content words and aligned word positions (Section 2.3).

(1) Estimate word keyness based on TF*IDF
(2) Estimate word keyness based on semantic PageRank
(3) Combine word keyness from TF*IDF and PageRank
(4) Output the resulting word keyness

Figure 2. Outline of the process used to train the *KEA* system.

2.2 Word Keyness Estimation

We attempt to evaluate significance levels for words that are expected to reflect their keyness in parallel articles. Our learning process is shown in Figure 2.

In the first stage, we estimate words' keyness in the article pair (Art_o, Art_s) based on TF*IDF. As inspired by (Nelken and Shieber, 2006), we view sentences in both Art_o and Art_s as documents, and define the sentence-based TF to indicate the existence of a word in an article sentence and the sentence-based IDF to be the reciprocal of the sentential appearance of a word. The TF*IDF keyness of a word w in sentence *Sent* is $\text{tfidf}(w|Sent)=\text{TF}(w|Sent)\times\text{IDF}(w|\{Sent\})$ where $\text{TF}(w|Sent)$ is active and set to 1 if *Sent* contains w (0 otherwise) and $\{Sent\}$ represents the set of the article sentences in Art_o and Art_s. Take the words "*gorillas*" and "*of*" of sentence 1 in Figure 1 for example. "*gorillas*" is in 5 original sentences and 5 simplified ones, while "*of*" has 4 sentential occurrences in Art_o and 4 in Art_s. Thus, $\text{tfidf}(\text{"}gorillas\text{"}|\text{sentence 1})$ is $1\times1/(5+5)=0.1$ and $\text{tfidf}(\text{"}of\text{"}|\text{sentence 1})$ is $1\times1/(4+4)=0.125$.

As one can speculate, TF*IDF penalizes frequent content words (e.g., "*gorillas*" assigned 0.1 compared to "*of*" assigned 0.125), but frequent content words are more likely to be key words and should receive more attention during sentence aligning. Therefore, we also turn to PageRank, a famous keyword extraction algorithm, to infer word significance and give better share of weights for essential words.

In the second stage of the learning algorithm (Step (2) in Figure 2), we estimate words' keyness in Art_o and Art_s based on PageRank, or specifically semantically motivated PageRank. Figure 3 shows the algorithm for deriving keyness scores for article words.

In Step (1) of the algorithm, we view the original article Art_o and its simplified counterpart Art_s as a whole, following the sentence-wise TF*IDF. Then we construct PageRank word graph for the article pair. The graph is represented by a v-by-v matrix **EW** where v is the vocabulary size. **EW** stores normalized edge weights for word w_i and w_j (Step (3) and (4)). Note that the graph is directional (pointing from w_i to w_j) and that edge weights are associated with words' co-occurrence counts satisfying window size *WS*.

procedure EstimateKeyness(Art_o,Art_s,*KwPrefs*,m,λ)
(1) Concatenate Art_o with Art_s into *Content*
//Construct word graph for PageRank
(2) $\mathbf{EW}_{v\times v}=0_{v\times v}$
  for each sentence *st* in *Content*
    for each word pair w_i, w_j in *st* where $i<j$ and $j-i\leq WS$
      if not IsContWord(w_i) and IsContWord(w_j)
(3a) $\mathbf{EW}[i,j]+=1\times m$
      esle
(3b) $\mathbf{EW}[i,j]+=1$
(4) normalize each row of **EW** to sum to 1
//Iterate for PageRank
(5) set $\mathbf{NS}_{v\times v}$ to a diagonal matrix with
      $\mathbf{NS}[i,i]=RD(w_i|Art_o,Art_s)$
(6) set $\mathbf{KP}_{1\times v}$ to [*KwPrefs*(w_1),…,*KwPrefs*(w_v)]
(7) initialize $\mathbf{KY}_{1\times v}$ to [1/v,1/ v, …,1/v]
  repeat
(8a) **KY**'=λ×**KY**×**EW**×**NS** + (1-λ)×**KP**
(8b) normalize **KY**' to sum to 1
(8c) update **KY** with **KY**' after the check of **KY** and **KY**'
  until *maxIter* or avgDifference(**KY**,**KY**')≤*smallDiff*
  return **KY**

Figure 3. Evaluating word keyness via semantic PageRank.

In this paper, we exploit semantic features of word nodes to make PageRank semantically aware. Three types of semantic information are used. First, we weight edges according to the parts-of-speech of the connecting word nodes via edge multiplier $m>1$. The weighting mechanism concerns content words and function words. If a word is a function word and its connecting outbound word is a content word (i.e., nouns, verbs, adjectives and adverbs), their edge weight is conceptually enlarged m times (Step (3a)), 1 otherwise (Step (3b)). The goal of this multiplier is to differentiate edges and increase the edge weights from function words to content words, which in turn propagates function words' PageRank scores more to content words and leads to content words' gains in importance.

The second semantic feature takes node's significance into account (Step (5)). Intuitively, if a word node is mentioned in Art_o as frequently as in Art_s, it is more likely to be an essential word, whereas if the degrees of reference of a word in Art_o and Art_s differ a lot, the word may not be as important. In our PageRank keyness estimation, a word node's reference distribution (i.e., *RD*) between Art_o and Art_s comes into play and is defined sentence-wise as

$$RD(w|Art_o, Art_s) = \frac{2\times \min\left(|\{Sent: Sent \in Art_o \cap w \text{ in } Sent\}|, |\{Sent: Sent \in Art_s \cap w \text{ in } Sent\}|\right)}{|\{Sent: Sent \in Art_o \cap w \text{ in } Sent\}| + |\{Sent: Sent \in Art_s \cap w \text{ in } Sent\}|}$$

where the numbers of sentences in Art_o and in Art_s containing the word w are leveraged. Take the word "*gorillas*" and "*army*" in Figure 1 for instance. "*gorillas*" occurs in Art_o as often as in Art_s while "*army*" only occurs in Art_o. As far as word keyness in sentence alignment concerns, "*gorillas*" is a much more significant word than "*army*", reflected by *RD*("*gorillas*")=1 being larger than *RD*("*army*").

We exploit the keyword preference model (i.e., **KP**) as the third semantic feature to distinguish words that tend to be keywords (Step (6)). TF*IDF scores of Step (1) in Figure 2 are used for this purpose and denoted by *KwPrefs*.

After Step (6) of Figure 3 sets the one-by-v matrix **KP**, Step (7) initializes the matrix **KY** of PageRank scores or, in our case, word keyness scores. Then, we re-distribute words' keyness scores until the number of iterations or the average score differences of two consecutive iterations reach their respective limits. In each iteration, a word's keyness score is the linear combination of its keyword preference score and the sum of the propagation of its inbound words' previous PageRank scores. And the sum of the propagation is further weighted by the word's degree of reference. Specifically, for the word w_j in *Content*, its PageRank score is computed as

$$\mathbf{KY'}[1,j]=\lambda\times(\textstyle\sum_{i\in v}\mathbf{KY}[1,i]\times\mathbf{EW}[i,j]\times\mathbf{NS}[j,j])+(1-\lambda)\times\mathbf{KP}[1,j]$$

where λ is referred as damping factor and usually set to 0.85. After the iterative process stops, the algorithm returns the scores as PageRank-based word keyness estimation.

In the final stage of training (Step (3) in Figure 2), we combine word keyness scores from TF*IDF and semantic PageRank. Note that to gather solid word statistics all article sentences are lemmatized and shallowly parsed with part-of-speech information. Example word keyness scores are shown in Figure 1. Notice that the word "*gorillas*" clearly gains more attention in terms of significance in the articles, compared to its TF*IDF estimation alone.

2.3 Run-Time Sentence Alignment

Once the keyness scores for words are automatically learned, they are stored for run-time query. *KEA* then aligns sentences of given monolingual parallel articles using the procedure in Figure 4. We first segment the original article Art_o and its simplified counterpart Art_s into sentences (Step (1)). And we employ a global dynamic programming with deletion and insertion operation to identify the parallel sentences between the monolingual article pair that are translations of each other with different readability or targeted for different groups of audience (from Steps (2) to (6)).

```
procedure AlignSentences(Art_o,Art_s,KY,x,N)
(1a) Segment Art_o into sentences Sent_o,1,…, Sent_o,m
(1b) Segment Art_s into sentences Sent_s,1,…, Sent_s,n
//initialization for dynamic programming
(2)  initialize DP_(m+1)×(n+1)=0_(m+1)×(n+1)
//recurrence for dynamic programming
     for 1 ≤ i ≤m
      for 1≤ j ≤n
(3a)   (W_o,CW_o)=findWordAndContentWord(Sent_o,i)
(3b)   (W_s,CW_s)=findWordAndContentWord(Sent_s,j)
(3c)   CosSim=findCosSimBasedOnWP(W_o,W_s,KY)
(3d)   AlignedRatio_cw=findCWAlignedRatio(CW_o,CW_s)
(3e)   DP[i+1,j+1]=CosSim × AlignedRatio_cw
                   +max{DP[i,j],DP[i+1,j],DP[i,j+1]}
//backtracking for dynamic programming
(4)  AG_m×n=backtrack(DP)
//deletion operation for the global dynamic programming
     for any i where |{j|AG[i,j]==1}|>1
(5a)   AG[i,j]=0 if Sent_s,j is not in Sent_o,i's top x similar
     for any j where |{i|AG[i,j]==1}|>1
(5b)   AG[i,j]=0 if Sent_o,i is not in Sent_s,j's top x similar
//insertion operation for the global dynamic programming
     for any (Sent_o,i,Sent_s,j) in the top N similar
(6)    AG[i,j]=1 if CosSim(Sent_o,i,Sent_s,j) > threshold
     return {(i,j)|AG[i,j]==1}
```

Figure 4. Aligning sentences at run-time.

The algorithm initializes a (m+1)-by-(n+1) matrix **DP** to store the optimal sentence alignment score. Specifically, **DP**[i+1,j+1] records the best score for aligning sentences between $Sent_{o,1},\ldots,$ $Sent_{o,i}$ and $Sent_{s,1},\ldots,$ $Sent_{s,j}$ (Step 2) where $1\leq i\leq m$, the number of the sentences in Art_o, and $1\leq j\leq n$, the number of the sentences in Art_s. Step (3a) and Step (3b) finds the word vector $\mathbf{W}_o=\{w_o\}$ of sentence $Sent_o$ and $\mathbf{W}_s=\{w_s\}$ of $Sent_s$ respectively. The word vectors are then used to estimate cosine-based sentence similarity:

$$CosSim(Sent_o, Sent_s) = \frac{\sum_{w\in\{w_o\}\cap\{w_s\}} \mathrm{KY}^2[w]}{\sqrt{\sum_{w_o} \mathrm{KY}^2[w_o] \times \sum_{w_s} \mathrm{KY}^2[w_s]}}$$

The cosine similarity is equipped with the knowledge of word keyness (i.e., **KY**) learned from the articles as in Section 2.2. Compared to the frequent sentence re-structuring and re-ordering, re-ordering of words in sentences seldom happens in simplification. In other words, words in the original sentences will be translated or simplified in order. As a result, word vectors contain words' relative positions in sentences, $posi(w)/|Sent|$ where absolute word positions are divided by sentential word lengths. And words' keyness scores are weighted by (1-*diff*) to consider the effort or travel distance needed to align words in sentences where *diff* is the absolute difference of the aligned words' relative word positions. Take the second sentence "*They aren't real animals, but statues dressed up as people.*" in the original article and the seventh sentence "*People made them.*" in the simplified in Figure 1 for example. The keyness of their common word "*people*" will be penalized by (1-|11/12-1/4|) since long-distance word alignment should be discouraged. Note that Step (3c) implements this word position functionality to encourage short-distance word alignment and punctuations should re-set the absolute word position to accommodate splits of article sentences.

Intuitively, mapping content words in sentences is more important than mapping non-content words. Therefore, Step (3e) further weights sentence-level cosine similarity using the aligning ratio of content words in sentences, $AlignedRatio_{cw}$, computed as $2\times|\mathbf{CW}_o \cap \mathbf{CW}_s|/(|\mathbf{CW}_o|+|\mathbf{CW}_s|)$ where the size of the common content words is divided by the sum of the size of the individual sentential content-word set. To allow for word changes, sentences' *CosSim* will only be penalized by $AlignedRatio_{cw}$ if their aligned content word ratio is below certain degree, which discourages the alignment of these sentences. Otherwise, *CosSim* will be left as it is.

Following (Gale and Church (1991)) and (Nelken and Shieber, 2006), Step (3e) computes the optimal alignment score for aligning $Sent_{o,1}$,…, $Sent_{o,i}$ and $Sent_{s,1}$,…, $Sent_{s,j}$ in global alignment

dynamic programming. The optimal score is recursively hypothesized to come from **DP**[*i*,*j*], **DP**[*i*+1,*j*], and **DP**[*i*,*j*+1]. Step (4) backtracks and returns an **AG** matrix where **AG**[*i*,*j*] is on if the best sentence aligning result contains $Sent_{o,i}$ and $Sent_{s,j}$ pair.

Subsequently, we prune the complete path by discarding sentence pair ($Sent_{o,i}$,$Sent_{s,j}$) whenever $Sent_{o,i}$ (or $Sent_{s,j}$) has multiple alignments and $Sent_{s,j}$ (or $Sent_{o,i}$) is not in $Sent_{o,i}$'s (or $Sent_{s,j}$'s) top *x* similar sentences in Step (5a) (or Step (5b)). *x* is used to control the one-to-many and many-to-one alignments. For instance, if *x* is set to two, the algorithm only allows each original sentence to be split to two simplified sentences and vice versa.

On the other hand, since the gaps and re-orderings between sentence alignments are more prominent in monolingual setting than in bilingual, Step (6) is to recover some of the missing aligning points in the optimal complete path and acts as a straightforward insertion operation. It activates **AG**[*i*,*j*] if ($Sent_{o,i}$,$Sent_{s,j}$) is one of the *N* most similar sentence pairs among the *m*×*n* sentence pairs and its similarity exceeds a certain threshold.

Once the complete path has been constrained to 1-to-*x* and *x*-to-1 purer alignments and expanded by high-confident alignments, the aligning points are returned as the final result produced by the *KEA* system. An example sentence alignment for monolingual parallel articles on our working prototype is shown in Figure 1.

3 Experiments

KEA was designed to identify sentences that are likely to be parallel in monolingual article pairs. As such, *KEA* will be evaluated over alignment accuracy at sentence level. Since the goal of *KEA* is to leverage word significance in sentence alignment, different estimation strategies for word keyness will be compared. In this section, we first examine the parallel level of English Wikipedia and Simple English Wikipedia, the original-simplified article pairs commonly used by text simplification community (Section 3.1). Section 3.2 presents the details of training *KEA* for the evaluation. Finally, we report system performance with different settings concerning keyness estimation for words, aligned content word ratios in sentences, and offsets of relative aligned word positions in Section 3.3. Section 3.3 also shows the results of our pilot study as to the effect of our sentence-aligned parallel articles on language learning.

3.1 English and Simple Wikipedia

This section examines the parallel level of English Wikipedia (EW) and Simple English Wikipedia (SEW), a common article source for training simplification model (e.g., Zhu et al., 2010 and Woodsend and Lapata, 2011). We manage to see if articles on SEW are written based on their counterparts on EW and to see if articles on EW and SEW are actually translations of each other with different target audiences in mind where SEW with basic vocabulary and grammar aims for lay people.

With language links and image files from Wikipedia, we were able to find 183K article pairs between EW and SEW in October, 2013. To see their parallel-ity, we randomly chose 10 pairs and hand aligned them at sentence level. Table 1 summarizes the alignment result.

	# sent on EW (# sent aligned)	# sent on SEW (# sent aligned)
article pair 1	136(1)	2(1)
article pair 2	145(1)	7(1)
article pair 3	86(2)	16(2)
article pair 4	180(2)	6(3)
article pair 5	166(2)	12(4)
article pair 6	242(16)	53(16)
article pair 7	8(1)	4(1)
article pair 8	2(2)	2(2)
article pair 9	160(1)	3(1)
article pair 10	70(1)	1(1)

Table 1. Alignment results of the sampled EW and SEW article pairs.

In Table 1 we list the numbers of sentences in articles on EW and SEW and enclose in parentheses the number of sentences that are manually aligned to its SEW or EW counterparts. We observe that (1) the numbers of sentences of the EW and SEW article pairs vary a lot; (2) only a handful of sentences in EW articles are aligned to, or kept in, SEW sentences; (3) most of time, alignments happen only at the first few article sentences except for the identical EW and SEW articles in article pair 8 and the much more parallel article pair 6. Surprisingly, these article pairs may not be as parallel as one may think, and SEW

articles are typically written on their own without referring to or seldom based on their EW counterparts.

Since our goal is to find sentence pairs in parallel articles which differ in readability, using English Wikipedia and Simple English Wikipedia may not be a good idea. Fortunately, there are monolingual *parallel* article pairs on the Web.

3.2 Training KEA

Based on the findings in Section 3.1, we collected (original) articles and their direct simplified counterparts, i.e., parallel articles, on the Web. English articles on websites NewsInLevels, BreakingNewsEnglish, and EasierEnglishWiki made up of our monolingual parallel corpus. These sites publish parallel news articles on daily or monthly basis and our current collection contains 607K words on the original side and 510K words on simplified.

100 article pairs were set aside and manually aligned for sentence alignment evaluation. This test set had 1,098 original and 1,285 simplified sentences. Specifically, there were 17K words in the testing original articles while there were 14K words in the simplified. Note that both training and testing article pairs were lemmatized and part-of-speech tagged by GENIA tagger from Tsujii lab (Tsuruoka and Tsujii, 2005).

3.3 Evaluation Results

In this section, we examine the effectiveness of *KEA*'s keyword-based weighting for aligning words, content-word alignment ratio, and offsets of relative aligning word positions, in monolingual sentence alignment (See Table 2).

	Precision	F-measure
KEA	85	83.9
KEA-WP	84.7	83.8
KEA-CW	83.4	83.1
TF*IDF	80	81.4

Table 2. Alignment performance (%).

Applied on monolingual parallel corpora, *KEA* with full capability outperforms the current state-of-the-art TF*IDF (Nelken and Shieber, 2006). Specifically, *KEA* further improves precision and f-measure relatively by 6.25% and 3%. Figure 5 shows a testing article pair's sentence alignment results done by TF*IDF and *KEA*. As we can see, although TF*IDF is a straightforward context-sensitive approach, it does not handle well with the two-word alignment between original sentence 1 and simplified sentence 1 (i.e., aligned words are "*mobile*" and "*phone*") and the two-word alignment between original sentence 3 and simplified sentence 2 (i.e., aligned words are "*they*" and "*with*"). By assigning keyword-based weights to words, *KEA* better distinguishes the importance of aligning "*mobile*" and "*phone*", and that of "*they*" and "*with*", in sentence pairs, and successfully identifies the alignment of (1,s1) and discards the alignment of (3,s2).

Original article:
(1) Mobile phones don't always work perfectly. (2) They can have a bad signal or a dying battery and they can make us very angry. (3) In Finland 12 years ago, they came up with a new idea. (4) They started to throw their phones as far as possible not only to make themselves feel better but also in the name of sports. (5) People from all over the world met for this year's event and one man from Finland threw his mobile phone 101 metres. (6) He was the winner. (7) He didn't practise much before the event. (8) He spent the day before in the pub.

Simplified article:
(s1) Mobile phones have sometimes problems. (s2) They have a bad signal or a bad battery and we are not happy with them. (s3) In Finland 12 years ago, they had a new idea. (s4) They started to throw their mobile phones. (s5) They tried to throw the phones very far. (s6) People from many countries met this year again. (s7) They threw the mobile phones again. (s8) The best man was from Finland. (s9) He threw his mobile phone 101 metres. (s10) He didn't train for this moment because he was in the pub.

(a) Alignments by TF*IDF: (2,s2), (3,s2), (3,s3), (4,s4), (4,s5), (5,s6), (5,s9), (7,s10), (8,s10)

(b) Alignments by *KEA*: (1,s1), (2,s2), (3,s3), (4,s4), (4,s5), (5,s6), (5,s9), (7,s10), (8,s10)

Figure 5. Alignment results of a testing article pair done by (a) TF*IDF (b) *KEA*.

In addition, Table 2 indicates that differences of relative positions of aligned words (i.e., *KEA* minus WP) and percentages of aligned content words with flexibility of vocabulary change (i.e., *KEA* minus CW) both plays a role in aligning sentences. Content-word alignment ratio, clearly, is a much more important feature in boosting alignment accuracy.

A pilot study, on the other hand, was conducted to see if monolingual parallel articles aligned at sentence level can help readers understand original

articles better than given article pairs with different readability. In this study, an English professor was asked to set multiple-choice reading comprehension exam paper for two of our testing article pairs. And a class of 16 college students learning English as a second language participated and was divided into two groups: one reading original articles and their simplified counterparts (i.e., control group) and the other reading the sentence-aligned article pairs (i.e., experimental group). Promisingly, our sentence alignment information helps the language learners. The experimental group outperforms the control relatively by 27.5% (51% vs. 40%) in reading comprehension test. Also, post-experiment survey indicates 85% of the participants found our sentence-aligned article pairs helpful in understanding the original or difficult articles.

Overall, we are modest to say that *KEA* can extract parallel sentences from monolingual articles more accurately than the current state-of-the-art, by identifying key words for alignment, and that *KEA* can yield original-to-simplified sentence pairs that are beneficial to language learners in article understanding or language learning.

4 Related Work

Sentence alignment has been regarded as an important first step for bilingual translation or monolingual translation/simplification. In our work we address an aspect of monolingual sentence alignment. More specifically, we focus on the first part of text simplification (Siddharthan, 2010; Zhu et al., 2010; Woodsend and Lapata, 2011; Biran et al., 2011), namely monolingual sentence alignment (MSA) on parallel articles.

The research in MSA starts in summarization. For example, Marcu (1999) leverages cosine measure to estimate sentence similarity while Jing (2002) uses Hidden Markov Model for sentence and summary matching. Hatzivassiloglou et al's SimFinder (1999; 2001), on the other hand, exploits word overlap and matching nouns to align sentences in multi-document summary.

Recent work has been using context information in MSA. Barzilay and Elhadad (2003) exploit inter-document topical sub-structures in Encyclopedia entries. Nelken and Shieber (2006) describe how to use sentence-based TF*IDF to weight aligned words. And their work has been suggested as the current state-of-the-art monolingual sentence aligner (Nelken and Shieber, 2006; Zhu et al., 2010).

In contrast to the previous research, we consider word keyness in aligning words during sentence alignment. The famous keyword extraction algorithm, PageRank (Mihalcea and Tarau, 2004; Padmanabhan et al., 2005; Liu et al., 2010; Zhao et al., 2011), is used to weight words and to favor the aligning of essential words in sentences. Word keyness, weighted by ratios of aligned content words and offsets of aligned relative word positions, is integrated into a global dynamic programming to identify parallel sentences in monolingual articles.

5 Summary and Future Work

We have introduced a method for learning to differentiate key words in sentence alignment on monolingual parallel articles, the very first step for text simplification. The method involves estimating word keyness based on TF*IDF and semantic PageRank, weighting keyword-based sentence-level cosine similarity via percentages of content word alignment and differences of relative positions of aligned word, and identifying parallel sentences using a global dynamic programming with deletion and insertion operations. We have implemented and evaluated the method as applied to monolingual sentence alignment and language learning. In the evaluation, we have shown that the method outperforms the current state-of-the-art in both alignment accuracy and f-measure, and that language learners benefit from our sentence-aligned monolingual parallel articles in reading comprehension test.

Many avenues exist for future research and improvement of our system. For example, we would like to see if we can boost simplification systems' performance using our better-aligned parallel sentences. And we would like to examine the possibility of employing such keyword concept to determine articles' good simplified versions. Yet another interesting direction to explore is to fully examine the possibility of using our aligned original and simplified sentence pairs for educational purposes.

Acknowledgement

This study is conducted under the "Online and Offline integrated Smart Commerce Platform (1/4)" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

References

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the EMNLP*.

Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the ACL*, pages 496-501.

William Gale and Kenneth Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the ACL*.

Vasileios Hatzivassiloglou, Judith Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. In *Proceedings of the EMNLP*.

Vasileios Hatzivassiloglou, Judith Klavans, Melissa Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. 2001. SIMFINDER: a flexible clustering tool for summarization. In *Proceedings of the Workshop on Automatic Summarization*, pages 41-49.

Hongyan Jing. 2002. Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527-543.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the EMNLP*, pages 366-376.

Daniel Marcu. 1999. The automatic construction of large-scale corpora for summarization research. In *Proceedings of the SIGIR*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing orders into texts. In *Proceedings of the EMNLP*, pages 404-411.

Rani Nelken and Stuart M. Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the EACL*, pages 161-168.

Divya Padmanabhan, Prasanna Desikan, Jaideep Srivastava, and Kashif Riaz. 2005. WICER: a weighted inter-cluster edge ranking for clustered graphs. In *Proceedings of the IEEE/WIC/ACM WI*, pages 522-528.

Advaith Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings of the INLG*, pages 125-133.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the EMNLP*, pages 467-474.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the EMNLP*, pages 409-420.

Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyword extraction from Twitter. In *Proceedings of the ACL*, pages 379-388.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the Coling*, pages 1353-1361.